CLAIMS

1. A method for detecting transfer errors in an address bus, comprising:

generating a first address parity using a memory address;

scrambling at least two data error-correction-code (ECC) check bits using the first address parity; and

writing the data ECC check bits to a memory, the data ECC check bits enabling detection of transfer errors in the address bus.

2. The method of claim 1, further comprising:

generating a second address parity using the memory address;

reading the data ECC check bits from the memory; and

unscrambling the at least two data ECC check bits using the second address parity, the data ECC check bits enabling detection of transfer errors in the address bus.

3. The method of claim 2, further comprising:

executing an ECC operation; and

reporting an ECC error to an exception-handling software if the at least two data ECC check bits signal an error.

4. The method of claim 2, wherein the method operation of unscrambling the at least two data ECC check bits is selected from one of using an exclusive-OR function and using an exclusive-NOR function.

5. The method of claim 1, wherein the method operation of unscrambling the at least two data ECC check bits is selected from one of using an exclusive-OR function and using an exclusive-NOR function.

6. The method of claim 1, wherein the at least two data ECC check bits are selected from the group consisting of two most significant bits and two least significant bits.

7. A method for detecting transfer errors in an address bus, comprising:

generating a second address parity using a memory address;

reading data error-correction-code (ECC) check bits from the memory; and

unscrambling at least two previously scrambled data ECC check bits using the second address parity, the data ECC check bits enabling detection of transfer errors in the address bus.

8. The method of claim 7, further comprising:

executing an ECC operation; and

reporting an ECC error to an exception-handling software if the at least two previously scrambled data ECC check bits signal an error.

9. The method of claim 7, wherein the method operation of unscrambling the at least two previously scrambled data ECC check bits is selected from one of using an exclusive-OR function and using an exclusive-NOR function.

10. The method of claim 7, wherein the at least two previously scrambled data ECC check bits are selected from the group consisting of two most significant bits and two least significant bits.

11. A system for detecting transfer errors in an address bus, comprising:

a first parity generator for generating a first address parity using a memory address; and

at least two exclusive-OR operators coupled to the first parity generator for scrambling at least two data error-correction-code (ECC) check bits using the first address parity,

wherein the first parity generator and the at least two exclusive-OR operators coupled to the first parity generator enable detection of transfer errors in the address bus.

12. The system of claim 11, further comprising:

a second parity generator for generating a second address parity using the memory address; and

at least two exclusive-OR operators coupled to the second parity generator for unscrambling the at least two data ECC check bits using the second address parity,

wherein the second parity generator and the at least two exclusive-OR operators coupled to the second parity generator enable detection of transfer errors in the address bus.

13. The system of claim 12, wherein the second parity generator and the at least two exclusive-OR operators coupled to the second parity generator are located in a memory controller, the memory controller being coupled to a memory by the address bus and a data bus.

14. The system of claim 13, wherein the memory controller is coupled to a central processing unit (CPU) core.

15. The system of claim 11, wherein the first and second parity generators and the at least two exclusive-OR operators coupled to the first and second parity generators are located in a memory controller, the memory controller being coupled to a memory by the address bus and a data bus.

16. The system of claim 11, wherein the first parity generator and the at least two exclusive-OR operators coupled to the first parity generator are located in a memory, the memory being coupled to a memory controller by the address bus and a data bus.

5

17. The system of claim 16, wherein the memory is selected from the group consisting of a static random access memory and a dynamic random access memory.